REMARKS

Applicant respectfully requests reconsideration and allowance of the subject application. Claims 1-15 and 22-32 are pending, of which claim 11 has been amended. The amendment to claim 11 is purely of form to correct informalities noted by the Applicant, and are not to overcome prior art or any other objections..

35 U.S.C. §103

Claims 1-15 and 22 are rejected under 35 U.S.C. §103(a) for obviousness over U.S. Patent No. 6,012,083 to Savitzky et al. (hereinafter, "Savitzky"), in view of U.S. Patent No. 5,956,483 to Grate et al. (hereinafter, "Grate") (*Office Action* p.2). It is noted that Microsoft is the owner of both the present application and Grate.

Savitzky describes a third-party "agency" computing system that is interposed between one or more Web clients and one or more Web servers to interact with the clients and servers to transfer documents (col. 3, lines 32-37). The Web clients and Web servers communicate documents via the agency with HTTP (hypertext transfer protocol) over a communication channel, such as the Internet (col. 5, lines 6-12).

Savitzky refers to applets, stating that "client-side code execution is limited to documents in which a server has included applets and is limited to use with applet-aware browsers" (col. 2, lines 41-43). This is essentially described in the "Background" section of Applicant's Specification and is an example of the very prior art that Applicant sought to overcome. Applets, by their very nature, raise security issues for local computer systems. In general, computer programs can be configured to cause harm to the local computer system. (*Specification* p.2, lines 24-26). Applet viewers prevent harm from execution of an applet. For example, applets are prevented by an applet viewer from writing data to any persistent storage, thus protecting current contents of the persistent storage. (*Specification* p.3, lines 2-7). A disadvantage of the isolation of applets is that other computer processes executing concurrently with and independently of the applet viewer cannot communicate with the applets. (*Specification* p.3, line 28 - p.4, line 1).

Grate describes a method for embedding client-side function calls within HTML (hypertext markup language) content such that a user can initiate an embedded function call by clicking on a corresponding button or link while viewing a document with a standard Web browser (col. 3, lines 13-22). Web function calling protocols are embodied within client and server software components which provide for the exchange of information between Web users and online merchants over the Internet (col. 3, lines 37-42).

Contrary to Savitzky and Grate, Applicant claims receiving a request for a document from an applet, where the request specifies a function, the execution of which performs a task that is unrelated to retrieval of any document specified in the request (*see* claim 1, for example). Applicant describes an interprocess communication mechanism in which applets can receive and respond to processing requests of other computer processes, and which can send processing requests to such other computer processes without requiring modification of applet viewers. Additionally, computer system security is preserved with interprocess communication because an applet is denied direct access to computer system resources. (*Specification* p.5, lines 24-25).

> **Claim 1** recites a method for serving remote procedure calls from an applet which executes within an applet viewer which in turn executes in a computer system that is serving said remote procedure calls, the method comprising:
>
> receiving from the applet which executes in the same computer system that serves said remote procedure calls, a request for a document according to a document retrieval protocol implemented on a computer network;
>
> determining that the request specifies a function which is defined within a computer process executing independently of the applet and applet viewer and which includes one or more computer instructions, execution of which performs a task which is unrelated to retrieval of any document specified in the request; and
>
> executing the function in the same computer system that is executing said applet and applet viewer to thereby cause execution of the one or more computer instructions in response to receipt of the request.

Savitzky and/or Grate do not teach or suggest the combination of elements recited in claim 1. Both Savitzky and Grate describe client *and* server systems communicating information between the systems via the Internet. However, claim 1 recites "an applet which executes within an applet viewer which in turn executes in a computer system", "the applet which executes in the same computer system", and "executing the function in the same computer system that is executing said applet and applet viewer." Neither Savitzky nor Grate teach or suggest the combination of elements recited in claim 1 in a "same computer system" environment.

Furthermore, Savitzky does not teach or suggest both a request for a document *and* "determining that the request specifies a function..., execution of which performs a task which is unrelated to retrieval of any document specified in the request", as recited in claim 1.

The Office suggests that Savitzky at col. 1, line 63 through col. 2, line 43 teaches the elements of claim 1 (*Office Action* pp.2-3). However, the cited section of Savitzky describes examples of server-client communications that teach away from Applicant's claim 1. For example, the Office cites that Savitzky describes a client sending a document request to a server for a document in the form of a URL that refers to a program on the server (*Savitzky* col. 2, lines 1-5). The Office disregards, however, that Savitzky continues the description with "[t]he server generates a document in accordance with the program and returns that document to the browser." (*Savitzky* col. 2, lines 5-7). This is expressly contrary to the execution of a function "which performs a task which is *unrelated* to retrieval of any document specified in the request", as recited in claim 1. To return a document to a client browser, the document request of Savitzky would be related to the retrieval of the document.

With regards to "applets", Savitzky describes that "[w]ith client-side code execution, the client requests a document and the returned document contains program code embedded in the document ..." which can be used for such tasks as animating graphic elements of a document (*Savitzky* col. 2, lines 25-31). This is also expressly contrary to "a request for a document" and "determining that the request specifies a function..., execution of which performs a task which is *unrelated* to retrieval of any document specified in the request", as positively recited in claim 1.

The Office states that it would be obvious that the applet generates the request since the request for a document is generated with client-side code execution and that "the script execution is unrelated to retrieval of the document request generated by the script" (*Office Action* p.3). Applicant respectfully disagrees. Savitzky clearly describes that "the server generates a document in accordance with the program and returns that document to the browser" (*Savitzky* col. 2, lines 5-7), and that with applets, the client requests a document and the returned document contains program code embedded in the document" (*Savitzky* col. 2, lines 25-27).

Grate also does not teach both a request for a document *and* "determining that the request specifies a function..., execution of which performs a task which is unrelated to retrieval of any document specified in the request", as recited in claim 1. Grate says nothing about calling or requesting a function with a request for a document having an encoded remote procedure calling request, as claimed by the Applicant.

Accordingly, claim 1 is allowable over the Savitzky-Grate combination and Applicant respectfully requests that the §103 rejection be withdrawn.

Claims 2-5 and 22 are allowable by virtue of their dependency upon claim 1. Additionally, claims 2, 3, and 4 are allowable for over the Savitzky-Grate combination for independent reasons.

Claim 2 recites "determining that the request includes a document specification which is in a portion of a name space reserved for function requests." The Office has not cited to either Savitzky or Grate for teaching "a name space reserved for function requests", or "a document specification which is in a portion of a name space reserved for function requests."

The Office states that it would be obvious that since the request is in a URL with parameters for the script execution, it can also contain how to format the returned document (*Office Action* p.4). Applicant respectfully disagrees that the features of claim 2 are obvious, and claim 2 does not recite anything about a format for a returned document. Furthermore, the Office has not provided any indication of such features in either Savitzky or Grate. Without some indication as to the basis for the rejection, Applicant is unable to formulate a detailed response. Accordingly, Applicant respectfully requests that the Office withdraw the §103 rejection of claim 2.

Claims 3 and 4 recite "returning to the applet result data produced by execution of the function" (claim 3), and "forming a document which includes the data and sending the document to the applet" (claim 4).

The additional elements recited in claim 4 is that result data produced by execution of the function (of claim 1) is included into a document and the document is sent to the applet. Neither Savitzky nor Grate teaches "forming a document which includes the data", and "sending the document to the applet", as recited in claim 4.

The Office suggests that Savitzky at col. 2, lines 10-14, teaches forming a document which includes the data and sending the document to the applet in a "dynamic document of server side code execution" (*Office Action* p.4). Applicant respectfully disagrees that the document of Savitzky includes the results *of a function*, execution of which performs a task which is unrelated to retrieval of any document specified in the request, as recited in the combination of claims 1, 3, and 4. Accordingly, claims 3 and 4 are allowable over the Savitzky-Grate combination.

<u>Claim 6</u> recites "receiving from the applet ... a request for a document", and "determining that the request specifies a function which is defined within a computer process executing independently of the applet and applet viewer and which includes one or more selected computer instructions, execution of which performs a task which is unrelated to retrieval of any document specified in the request."

As described above in the response to the rejection of claim 1, Savitzky and/or Grate do not teach or suggest both a request for a document *and* "determining that the request specifies a function which is defined within a computer process executing independently of the applet and applet viewer..., execution of which performs a task which is unrelated to retrieval of any document specified in the request", as recited in claim 6.

Accordingly, claim 6 is allowable over the Savitzky-Grate combination and Applicant respectfully requests that the §103 rejection be withdrawn.

<u>Claims 7-10</u> are allowable by virtue of their dependency upon claim 6. Additionally, claims 7, 8, and 9 are allowable for over the Savitzky-Grate combination for independent reasons.

<u>Claim 7</u> recites "determining that the request includes a document specification which is in a portion of a name space reserved for function requests."

As described above in the response to the rejection of claim 2, The Office has not cited to either Savitzky or Grate for teaching "a name space reserved for function requests", or "a document specification which is in a portion of a name space reserved for function requests." Accordingly, Applicant respectfully requests that the Office withdraw the §103 rejection of claim 7.

Claims 8 and 9 recite "returning to the applet result data produced by execution of the function" (claim 8), and "forming a document which includes the result data and sending the document to the applet" (claim 9).

As described above in the response to the rejection of claims 3 and 4, Neither Savitzky nor Grate teaches "forming a document which includes the result data", and "sending the document to the applet", as recited in claim 9. The document of Savitzky does not include the results *of a function*, execution of which performs a task which is unrelated to retrieval of any document specified in the request, as recited in the combination of claims 6, 8, and 9. Accordingly, claims 8 and 9 are allowable over the Savitzky-Grate combination.

Claim 11 recites "receiving from the applet ... a request for a document" and "determining that the request specifies a function which is defined within the computer process and which includes one or more computer instructions, execution of which performs a task which is unrelated to retrieval of any document specified in the request."

As described above in the response to the rejection of claim 1, Savitzky and/or Grate do not teach or suggest both a request for a document *and* "determining that the request specifies a function which is defined within the computer process and which includes one or more computer instructions, execution of which performs a task which is unrelated to retrieval of any document specified in the request", as recited in claim 11.

Accordingly, claim 11 is allowable over the Savitzky-Grate combination and Applicant respectfully requests that the §103 rejection be withdrawn.

Claims 12-15 are allowable by virtue of their dependency upon claim 11. Additionally, claims 12, 13, and 14 are allowable for over the Savitzky-Grate combination for independent reasons.

Claim 12 recites "determining that the request includes a document specification which is in a portion of a name space reserved for function requests."

As described above in the response to the rejection of claim 2, The Office has not cited to either Savitzky or Grate for teaching "a name space reserved for function requests", or "a document specification which is in a portion of a name space reserved for function requests." Accordingly, Applicant respectfully requests that the Office withdraw the §103 rejection of claim 12.

Claims 13 and 14 recite "returning to the applet result data produced by execution of the function" (claim 13), and "forming a document which includes the result data and sending the document to the applet" (claim 14).

As described above in the response to the rejection of claims 3 and 4, Neither Savitzky nor Grate teaches "forming a document which includes the result data", and "sending the document to the applet", as recited in claim 14. The document of Savitzky does not include the results *of a function*, execution of which performs a task which is unrelated to retrieval of any document specified in the request, as recited in the combination of claims 11, 13, and 14. Accordingly, claims 13 and 14 are allowable over the Savitzky-Grate combination.

New Claims

New claims 23-32 are presented for examination. Based on the above discussion regarding Savitzky and Grate, Applicant believes that claims 23-32 are also allowable. Support for claims 23-32 can be found in the Figures and in the Specification (see for example, pages 3, 5, and 7-8).

Conclusion

Pending claims 1-15 and 22-32 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. If any issues remain that preclude issuance of this application, the Examiner is urged to contact the undersigned attorney before issuing a subsequent Action.

Respectfully Submitted,

Dated: Feb 7, 2002

By: Dave Morasch
David A. Morasch
Reg. No. 42,905
(509) 324-9256 ext. 210

Version of amended specification with markings to show changes made

The paragraph starting at page 8, line 20, is amended as follows:

RPC process 210 and applet 200 are described in greater detail in the context of Figure 2. Applet 200 is configured to invoke RPC functions, e.g., either of RPC functions 206A-B of RPC process 210, to thereby incorporate the tasks performed by such RPC functions into a larger task performed by applet 200. RPC process 210 includes an HTTP server 204 which serves HTTP requests in a conventional manner, i.e., receives a URL which specifies a requested document and produces the requested document in response to the received URL. RPC process 210 also includes a URL filter 202 and RPC functions 206A-B. URL filter 202 reserves a portion of the name space of documents which can be requested using a URL for RPC requests. As described more completely below, URL filter 202 determines whether a particular URL specifies a document in the reserved name space portion and processes the URL ~~according~~ accordingly. In accordance with HTTP, applet 200 sends a URL specifying a document to RPC process 210 and receives the specified document from RPC process 210. To invoke either of RPC functions 206A-B, applet 200 forms a URL according to the steps of logic flow diagram 300 (Figure 3) and sends the URL to RPC process 210.

The paragraph starting at page 12, line 17, is amended as follows:

Applet 200 (Figure 2) can make itself available to receive RPC requests from RPC process 210 in a manner which is generally permitted by applet viewer 150 (Figure 1) and which is illustrated in logic flow diagram 700 ~~(~~(Figure 7). Processing according to logic flow diagram 700 begins in step 702.

The paragraph starting at page 12, line 21, is amended as follows:

In step 702, applet 200 ~~(~~(Figure 2) builds an RPC request for execution of an "RPC ready" RPC function by RPC process 210 and encodes the RPC request as a URL in the manner described more completely above. In step 704 ~~(~~(Figure 7), applet 200 ~~(~~(Figure 2) sends the URL encoded in step 702 ~~(~~(Figure 7) to RPC process 210 to thereby request execution of the "RPC ready" RPC function, which can be RPC function 206B, for example.

The paragraph starting at page 12, line 26, is amended as follows:

The design and implementation of RPC function 206B is such that execution thereof indicates to RPC process 210 that applet 200 is ready to receive RPC requests from RPC process 210 and establishes a communications channel through which RPC process 210 can send RPC requests to applet 200. Specifically, HTTP, as implemented by both RPC process 210 and applet viewer 150 ~~(~~(Figure 1) within which applet 200 executes, expects a document to be retrieved in response to the URL sent in step 704 ~~(~~(Figure 7 4). In addition, HTTP as implemented permits transfer of the requested document to be delayed and intermittent. However, ~~to~~ RPC process 210 requests a virtual document, i.e., a document which does not exist within memory 104 ~~(~~(Figure 1) of computer system 100 but which is instead created in response to the URL. Execution of RPC function 206B ~~(~~(Figure 2) of RPC process 210 changes the state of RPC process 210 to indicate that applet 200 is ready to receive RPC requests and to store data identifying the communications channel through which applet 200 is waiting to receive a document in response to the URL sent in step 704 ~~(~~(Figure 7).

The paragraph starting at page 13, line 10, is amended as follows:

RPC process 210 includes a core function 208 which defines and implements a central task for which RPC process 210 is designed. Execution of core function 208 can include sub-tasks which are implemented by one or more of RPC functions 212 of applet 200. Accordingly, to cause performance of such sub-tasks, core function 208 of RPC process 210 builds RPC requests which request execution of a selected one of RPC functions 212 and includes zero or more parameters to be used by the selected RPC function as input data. To send such an RPC request to applet 200, RPC process 210 sends the RPC request to applet 200 as a portion of the virtual document requested by the URL sent by applet 200 in step 704 ~~(~~(Figure 7). By sending the RPC request as only a portion of the requested virtual document, RPC process 210 ~~(~~(Figure 2) indicates to applet 200 that other RPC requests can be subsequently sent to applet 200 through the same communication channel. Since the RPC request is sent to applet 200 as part of a document, the contents of which are not constrained by any particular protocol such as HTTP, the RPC request can be in any convenient form and can be in a form which is entirely inappropriate for a HTTP URL. To terminate the communication channel, and therefore terminate the ability of applet 200 to receive RPC requests from RPC process 210, RPC process 210 sends data indicating that the entirety of the virtual document requested by applet 200 has been sent to applet 200.

The paragraph starting at page 13, line 26,is amended as follows:

Processing by applet 200 transfers from step 704 ~~(~~(Figure 7) to loop step 706 in which steps 708-712 are performed repeatedly until applet 200 ~~(~~(Figure 2) receives data indicating that the entirety of the requested virtual document has been received. In step 708 ~~(~~(Figure 7), applet 200 ~~(~~(Figure 2) receives a portion of the virtual document from RPC process 210. In step 710 ~~(~~(Figure 7), applet 200 ~~(~~(Figure 2) parses an RPC request from the received portion. As described above, the format of the RPC request can be entirely independent of the format of HTTP URLs. In step 712 ~~(~~(Figure 7), applet 200 ~~(~~(Figure 2) services the parsed RPC request by executing ~~the~~ one of RPC functions 212 specified by the parsed RPC request and supplying any arguments parsed from the received portion as input data. Any results produced by servicing the parsed RPC request can be communicated to RPC process 210 in the form of an HTTP URL built and sent to RPC process 210 in the manner described above. The results URL identifies the ~~one of~~ RPC functions 212 invoked by the parse RPC request to specify to RPC process 210 to which RPC request the resulting data pertains.

The paragraph starting at page 14, line 10, is amended as follows:

Steps 708-712 ~~(~~(Figure 7) are repeated until applet 200 ~~(~~(Figure 2) receives data from RPC process 210 indicating that the entirety of the requested virtual document has been sent to applet 200. Thereafter, processing according to logic flow diagram 700 completes.

The paragraph starting at page 14, line 13, is amended as follows:

In this way, applet 200 accepts RPC requests from RPC process 210 in a manner which is permitted by applet viewer 150 ~~(~~(Figure l) without requiring modification of applet viewer 150. Accordingly, many of the advantages of interprocess communication are achieved in the secure context of an applet viewer.

Version of amended claims with markings to show changes made

11. **(Amended)** A computer system comprising:

a processor;

a memory operatively coupled to the processor; and

a computer process which executes in the processor from the memory and which, when executed, serves remote procedure calls received from an applet which executes within an applet viewer which in turn executes in the processor from the memory concurrently and independently from the computer process, wherein the computer process serves the remote procedure calls by performing the steps of:

receiving from the applet which executes in the same computer system that serves remote procedure calls, a request for a document according to a document retrieval protocol implemented on a computer network;

determining that the request specifies a function which is defined within the computer process and which includes one or more computer instructions, execution of which performs a task which is unrelated to retrieval of any document specified in the request; and

executing the function in the same computer system that is executing said applet and applet viewer to thereby cause execution of the one or more computer instructions in response to receipt of the request.